Document Name:	AT137/139 TCP/IP APPNOTES
Version:	01.00
Date:	2010-11-18
Doc ID:	
Status:	Release

Copy right

This document contains proprietary information of ATWin. Copying or disclosing it to the third party is forbidden without the express authority. ATWin Ltd. reserves the right to make changes the information contained herein at any time without notice. The information herein is for reference only. ATWin Ltd. has no responsibility or liability for any errors or inaccuracies that may occur in this document.

Release Notes

Version	Date	Notes	Author	Remark
V1.0	2010-11-18	First Version	Steven Shao	

CONTENTS

The Embedded TCP/IP Stack

1 Introduction

1.1 Scope

This functional spec is intended to describe the TCP/IP protocol stack embedded in the AT137/139. It has 3 main contents:

- General introduction of TCP/IP stack in AT137/139.
- Description of the related AT commands and how to use them to acces the Internet.
- Introduction of the socket functions provided for TCP/IP applications.

1.2 Abbreviations and Terms

Specific abbreviations and terms used in this document are described here:

TE Terminal Endpoint

LWIP An entire TCP/IP protocol stack

PPP Point to Point Protocol

DNS Domain Name Server

GGSN Gate GPRS Support Node

ICMP Internet Control Message Protocol

2 Overview

AT137/139 implement an embedded TCP/IP stack. This stack can be directly driven by AT commands from the terminal equipment. The advantage of this solution is that it eliminates the need for the application manufacturer to implement their own TCP/IP and PPP stacks, thus minimizing cost and time to integrate Internet connectivity into a new or existing host application. The internal TCP/IP stack allows TCP connections are limited to a maximum of 4.

3 TCP/IP Related AT Commands

3.1 General Introduction

There are 11 new AT commands and 9 new URC used for TCP/IP features:

--**MIPCALL**: Internet Connect

AT+ MIPCALL Trigger the Internet connect and disconnect
AT+ MIPCALL =? Show the available choice of the prarameter.
AT+ MIPCALL? Show the current status of the Internet connection.

--**MIPOPEN**: Open a Socket (UDP or TCP)

AT+ MIPOPEN After connected with Internet ,initialize a new socket which could be a client or a server of TCP,or be the type of UDP.
AT+ MIPOPEN =? Show the available prarameter choice of the socket.
AT+ MIPOPEN? Show the current available sockets that can be opened

---

-- **MIPCLOSE**: Close a Socket (UDP or TCP)

AT+ MIPCLOSE Close a socket which have been opend
AT+ MIPCLOSE =? Show the available sockets.
AT+ MIPCLOSE? Show the current active sockets which have been opened

---

--**MIPSETS**: Set Size and Timeout for Automatic Push

AT+ MIPSETS After opening a socket, set size and timeout for automatic push data from buffer to Protocol Stack
AT+ MIPSETS =? Show the available prarameter choice of the socket.
AT+ MIPSETS? Show the current settings including size and timeout for automatic push of every socket

---

--**MIPSEND**: Send Data

AT+ MIPSEND After opening a socket ,send data to buffer
AT+ MIPSEND =? NOT SUPPORT
AT+ MIPSEND? Show the current free size of buffer of every socket

---

--**MIPTSEND**: Send data transparently

AT+ MIPTSEND After opening a socket ,send data to buffer
AT+ MIPTSEND =? Show the format and available parameters of this command.
AT+ MIPTSEND? Show the current opening sockets

--**MIPPUSH**: Push Data into Protocol Stack

AT+ MIPPUSH After sending data into buffer,push data from buffer into protocol stack

AT+ MIPPUSH =? Show the format and available parameters of this command.
AT+ MIPPUSH? Show the current opening sockets

--**MIPFLUSH**: Delete Data from Buffers

AT+ MIPFLUSH After sending data into buffer ,delete data accumulated in its accumulating buffers.
AT+ MIPFLUSH =? Show the current available socket.
AT+ MIPFLUSH? Show the current opening sockets

--**MIPCONF**: Configure Internal TCP/IP stack

AT+ MIPCONF Configure TCP stack parameters, such as retransmissions number, upper and bottom
limits of retransmission timeout and close delay
AT+ MIPCONF =? Show the format and available parameters of this command.
AT+ MIPCONF? Show the parameters of TCP/IP stack of every socket.

--**MPING**: Start Ping Execution

AT+ MPING Set parameters of ping execution and start ping execution
AT+ MPING =? Show the format and available parameters of this command.
AT+ MPING? Show the current parameters of ping execution

--**MSDNS**: Set DNS IP Address

AT+ MSDNS Set DNS IP address for each socket
AT+ MSDNS =? Show the format of this command.
AT+ MSDNS? Show DNS IP address of each socket

URC -- **MIPCALL**: Indicate Connection or Disconnection with the GGSN.

+ MIPCALL: This URC is sent by the MODEM to the terminal when the Modem connect or disconnect with the GGSN.

URC --**MIPOPEN**: Indicate Modem Open a new socket

+ MIPOPEN: This URC is sent by the MODEM to the terminal when the Modem initialize a new socket.

URC -- **MIPCLOSE**: Indicate Modem Open a new socket

+ MIPCLOSE: This URC is sent by the MODEM to the terminal when the Modem close a socket.

URC --**MIPSEND**: Indicate buffer Size of socket Change

+ MIPSEND: This URC is sent by the MODEM to the terminal when the buffer size changed .

URC --**MIPPUSH**: Indicate the data in buffer is pushed into the protocol stack.

+ MIPPUSH: This URC is sent by the MODEM to the terminal when the data in buffer is pushed into
the protocol stack.

URC --**MIPRTCP**: Indicate Receive Data from TCP Protocol Stack

---

+ MRTCP: This URC is sent by the MODEM to the terminal if MODEM receiving data from TCP Protocol Stack.

URC --**MIPRUDP**: Indicate Receive Data from UDP Protocol Stack

---

+ MRUDP: This URC is sent by the MODEM to the terminal if MODEM receiving data from UDP Protocol Stack.

URC --**MIPSTAT**: Socket Status Report

+ MIPSTAT: This URC is sent by the MODEM to the terminal if some socket closed

---

URC --**MPINGSTAT**: Status Update for +MPING Execution

+ MPINGSTAT: This URC is sent by the MODEM to the terminal after ping execution

3.2 AT+MIPCALL

3.2.1 set command

This command sets up a PPP connection with the GGSN, and returns a valid dynamic IP .

Syntax:

AT +MIPCALL=<Operation>[<APN>,[<User name>,<Password>]]

Response :

+MIPCALL : <status>

OK

Note:“MIPCALL” is a URC. This URC is sent by the MODEM to the terminal when the Modem connect or disconnect with the GGSN.

<Parameter> Description

<Operation>
0 Disconnect
1 Connect
<APN> Access Point Name, such as CMNET
<User name> User name in provider server (in quotation marks). Contact your service provider for details.
<Password> Password for provider server (in quotation marks). Contact your service provider for details.
<status>
0 Disconnected
1 Connected

Example:

AT+MIPCALL=1,"cmnet"
OK
+MIPCALL: 1 //connected
AT+MIPCALL=0
OK
+MIPCALL: 0 //disconnected

3.2.2 query command

Show the current status of the Internet connection.
Syntax:
AT+MIPCALL?

Response :
+MIPCALL: <status>[,<IP>]

Example:

AT+MIPCALL?
+MIPCALL: 1,10.72.73.24
OK
AT+MIPCALL?
+MIPCALL: 0
OK

3.2.3 test command

Show the current status of the Internet connection.
Syntax:
AT+MIPCALL=?

Response :
+MIPCALL : (list of supported <operation>s)
OK

Example:

AT+MIPCALL=?
+MIPCALL: (0,1)
OK

3.3 AT+MIPOPEN

3.3.1 set command

This command initialize a new socket that waits for a connection from a remote machine or opens a common or TCP secured with SSL connection with a remote side (according to received parameters). Each socket allocates an accumulating buffer whose size is 1460 bytes.
Syntax:
AT+MIPOPEN=<Socket_ID>,<Source_Port>,<Remote_IP>,<Remote_Port>,<Protocol>
Response:
+MIPOPEN: <Socket_ID>,<State>
Note: “MIPOPEN” is a URC. This URC is sent by the MODEM to the terminal when the Modem initialize a new socket.

<Parameter> Description
<Socket_ID> A unique number that identifies a connection.
Valid socket numbers - 1,2,3 and 4
<Source_Port> Port of source site.
Port range: 0-65535 (decimal digits)
<Remote_IP>
IP: IP of the remote site in the format "AAA.BBB.CCC.DDD". The range of each octet is 0-255. Value can be written in 1, 2, or 3 digits.
Host name: of remote site. The host-name convention should meet the rules as describe in RFC-1035 section: 2.3 Conventions. Syntax is not validated, except the maximum length (255 characters).

<Remote_Port> Port of source site.
Port range: 0-65535 (decimal digits)
Port 0 for incoming connection.
<Protocol> Type of protocol stack.
0 TCP
1 UDP
<State>
0 Inactive
1 Active
2 SSL secured

Example:

AT+MIPOPEN=1,5000,"www.sina.com",80,0
+MIPOPEN: 1,1 //socket open success
OK

3.3.2 query command

Show the current available sockets that can be opened
Syntax:
AT+MIPOPEN?
Response:
+MIPOPEN: <Socket_ID>
OK

Example:

AT+ MIPOPEN ?
+ MIPOPEN: 2,3,4 //socket 1 has been opened
OK

3.3.3 test command

Show the available prarameter choice of the socket.
Syntax:
AT+MIPOPEN=?
Response:
+MIPOPEN: (list of supported<socket_ID>s),(list of supported<source_port>s),(list of supported<"Destination_IP">s), (list of<destination_port>s),(list of supported <protocol>s)
OK

Example:

AT+MIPOPEN=?

+ MIPOPEN: (1-4),(0-65535),(<IP>),(0-65535),(0,1)

OK

3.4 AT+MIPCLOSE

3.4.1 set command

This command free the socket accumulating buffer and to close the socket.

Syntax:

AT+MIPCLOSE:<Socket_ID>

Response:

+MIPCLOSE:<Socket_ID>, [,<number_of_unacknowledged_bytes>],<close_type>

OK

Note: "MIPCLOSE" is a URC. This URC is sent by the MODEM to the terminal when the Modem close a socket.

Note: All data stored in the accumulating buffer will be lost.

<Parameter> Description

<Socket_ID> Unique number that identifies a connection.

Valid socket numbers - 1, 2, 3 and 4

<number_of_unacknowledged_bytes> Total number of bytes that were unacknowledged

<close_type>

0 - Connection was closed correctly.

1 – Socket closed error

Example:

AT+ MIPCLOSE=1

+ MIPCLOSE: 1,0 //socket close success

OK

3.4.2 query command

Show the current active sockets which have been opened

Syntax:

AT+MIPCLOSE?

Response:

+ MIPCLOSE: <Socket_ID>

Example:

AT+ MIPCLOSE?

+ MIPCLOSE: 1 //socket 1 is opened

OK

3.4.3 test command

Show the available sockets.

Syntax:

AT+MIPCLOSE=?

Response:

+MIPOPEN: (1-4)

OK

Example:

AT+ MIPCLOSE =?

+MIPCLOSE: (1-4)

OK

3.5 AT+MIPSETS

3.5.1 set command

This command to set a watermark in the accumulating buffer and set timeout after open a socket . When the watermark is reached, data is pushed from the accumulating buffer into the protocol stack.Arriving data to accumulating buffer triggers a start of time (defined in timeout) countdown. When counter reaches zero, data is moved into the protocol stack. If new data arrived before time is reached zero, it is re_initialized. If data in accumulating buffer reached watermark it is pushed to the accumulating buffer as usual, but if after automatic push there is some remaining data, time countdown is started.

Note: If there is data in the accumulating buffer, the +MIPSETS command will be rejected.

Syntax:

AT+MIPSETS=<Socket_ID>,<Size>[,<Timeout>]

Response:

+MIPSETS: 0 or 3 (0 means set success and 3 means Operation not allowed)

OK

<Parameter> Description

<Socket_ID> Unique number that identifies a connection.

<Size> Size of the buffer

1< size <= 1460, the default value is 1460.

<Timeout> 0 - 5000

0 means no timeout is used (default).

Example:

AT+ MIPSETS =1,30

+MIPSETS: 0 //socket parameters set success

OK

AT+ MIPSETS =3,12,12

+MIPSETS: 3 //there is data in buffer,set fail

OK

3.5.2 query command

Show the current settings including size and timeout for automatic push of every socket

Syntax:

AT+MIPSETS?

Response:

+MIPSETS: [<Socket_ID>,<Current Size Settings>,< Timeout>] For all ACTIVE sockets.

OK

Example:

AT+ MIPSETS =3,12

+MIPSETS: 0

OK

AT+ MIPSETS?

+MIPSETS: 1,1460,0 //socket one is opened and the setting parameters are default

+MIPSETS: 3,12,0 //the timeout is the default

OK

3.5.3 test command

Show the available prarameter choice of the socket.

Syntax:

AT+MIPSETS=?

Response:

+MIPSETS: (1-4),(list of supported<size>s),),(list of supported<Timeout>s)
OK

Example:

AT+ MIPSETS =?
+MIPSETS: (1-4),(1-1460),(0-5000)
OK

3.6 AT+MIPSEND

3.6.1 set command

This command send data into accumulating buffer which provided by the terminal . And when the amount of data reaches the predefined amount (see “+MIPSETS, Set Size and Timeout forAutomatic Push”) and then send this data using an existing protocol stack.Before sending data, a valid connection must be created using the+MIPCALL and +MIPOPEN commands. Recommends that the terminal sets the watermark in the accumulating buffer prior to this command, using the +MIPSETS command.
Note: “MIPSEND” is a URC. This URC is sent by the MODEM to the terminal when the buffer size changed .
Syntax:
+MIPSEND=<Socket_ID>,<Data>
Response:
+MIPSEND: <Socket_ID>,<Status>,<FreeSize>
OK

<Parameter> Description
<socket_ID> 1,2,3,4 Number of valid socket
<FreeSize> Free space in current buffer. Free size is calculated from the 1460.
0< Free Size < 1460
<Data> User data string is sent encoded with 0-F hexadecimal digits
<Status>
0 - Success
1 - Socket is flowed off

Example:

AT+ MIPSEND?
+MIPSEND: 3,1460
OK
AT+ MIPSEND =3,"121212"

+MIPSEND: 3,0,1457 //send success and the free size of the buffer is 1457
OK

3.6.2 query command

Show the current free size of buffer of every socket
Syntax:
AT+ MIPSEND?
Response:
+ MIPSEND: <Socket_ID>,<FreeSize> For all ACTIVE sockets
OK

Example:

AT+ MIPSEND?
+MIPSEND: 3,1456
+MIPSEND: 4,1460
OK

3.6.3 test command

NOT SUPPORT

3.7 AT+MIPTSEND

3.7.1 set command

This command send data into accumulating buffer and then send this data using an existing protocol stack.The maximum data length is 1460 bytes. Before sending data, a valid connection must be created using the+MIPCALL and +MIPOPEN commands.

Syntax:
+MIPTSEND=<Socket_ID>
> <Data> <ctrl+z>
Response:
+MIPPUSH: <Socket_ID>,<Status>,<Length>
OK

<Parameter> Description
<socket_ID> 1,2,3,4 Number of valid socket
<Length> Total sent data
<Data> User data need to be sent, hex and string accepted
<Status>
0 - Success
1 - Socket is flowed off

Example:

AT+ MIPTSEND = 1
> hello<ctrl+z>
+MIPPUSH: 1,0,5 // 5 Bytes is sent

3.7.2 query command

Show the current opening sockets
Syntax:
AT+ MIPTSEND?
Response:
+ MIPTSEND: <Socket_ID>
OK

Example:

AT+ MIPTSEND =?
+ MIPTSEND: 3 //socket 3 is opened
OK

3.7.3 test command

Show the current available socket
Syntax:
AT+ MIPTSEND=?
Response:
+ MIPTSEND: <Socket _ID>
OK

Example:

AT+ MIPTSEND =?

+MIPTSEND: (1-4)

OK

3.8 AT+MIPPUSH

3.8.1 set command

This command push the data accumulated in its accumulating buffers into the protocol stack. It is assumed that before using this command, some data should exist due to previous +MIPSEND commands.

Note : If the socket is a TCP socket , <"Destination_IP">,<Destination_Port> can be ignore, but if the socket is a UDP socket , <"Destination_IP">,<Destination_Port> is necessary.

Note: “MIPPUSH” is a URC. This URC is sent by the MODEM to the terminal when the data in buffer is pushed into the protocol stack.

Syntax:

AT+MIPPUSH=<Socket_ID>[,<"Destination_IP">,<Destination_Port>]

Response:

+ MIPPUSH: <Socket_ID>,<Status>[,<accumulated_sent_length>]

OK

<Parameter> Description

<Socket_ID> 1,2,3,4 Number of valid socket

<Destination_IP> IP of destination site in the format AAA.BBB.CCC.DDD. The value can be written in1, 2 or 3 digits. Destination_Port 0-65535 Port of destination site. Written in decimal digits.

<Status>

0 - Success

1 - socket is flowed off

2 - there is no data in socket to send

3 – socket ok,push error

<accumulated_sent_length> This parameter counts how many bytes were sent to the remote side from the TCP/IP stack. When user open socket, <accumulated_sent_length> initialized to Zero

Example:

AT+ MIPSEND =3,"121212"

+MIPSEND: 3,0,1457 //there is 3 byte data in the buffer
OK
AT+MIPPUSH=3,"218.18.42.76",5000
+MIPPUSH: 3,0 //push data into protocol stack success
OK

3.8.2 query command

Show the current opening sockets
Syntax:
AT+MIPPUSH?
Response:
+ MIPPUSH: <Socket_ID>
OK

Example:

AT+MIPPUSH=?
+MIPPUSH: 1 //socket 1 is opened
OK

3.8.3 test command

Show the format and available parameters of this command.
Syntax:
AT+MIPPUSH=?
Response:
+ MIPPUSH: <Socket _ID>,<IP>,<Port>
OK

Example:

AT+MIPPUSH=?
+MIPPUSH: (1-4),(<IP>),(1-65535)
OK

3.9 AT+MIPFLUSH

3.9.1 set command

This command delete data accumulated in its accumulating buffers after open a socket.
Note : If the socket is a TCP server, the command is invalid.
Syntax:
AT+MIPFLUSH = <Socket_ID>
Response:
+ MIPFLUSH: <Socket _ID>
OK

Example:

AT+ MIPSEND?
+MIPSEND: 1,1457 // there is 3 byte data in the buffer
OK
AT+ MIPFLUSH =1
+ MIPFLUSH: 1 //delete data of buffer success
OK
AT+ MIPSEND?
+MIPSEND: 1,1460 //there is no data in the buffer
OK

3.9.2 query command

Show the current opening sockets
Syntax:
AT+ MIPFLUSH?
Response:
+ MIPFLUSH: <Socket_ID>
OK

Example:

AT+ MIPFLUSH?
+ MIPFLUSH: 1 //socket 1 is opened
OK

3.9.3 test command

Show the current available socket
Syntax:
AT+ MIPFLUSH=?
Response:
+ MIPFLUSH: <Socket _ID>
OK

Example:

AT+ MIPFLUSH =?
+MIPPUSH: (1-4)
OK

3.10 AT+MIPCONF

3.10.1 set command

This command allows to configure TCP stack parameters, such as retransmissions number, upper and bottom limits of retransmission timeout, close delay. It can be used to configure TCP socket parameters before socket is opened .
Syntax:
AT+MIPCONF=<socket_ID>[,[<retr_num>],[<min_TO>],[<max_TO>],[<max_close_dela>],[,<is_nack_ind_r
eq>]]
Response:
OK

<Parameter> Description
<socket_ID> Number of configured TCP socket (1 to 4)
<retr_num> Number of retransmissions (1 to 5),the default value is 4.
<min_TO> Bottom limit to retransmit timeout (100 ms to 1 sec.),the default value is 5.
<max_TO> Upper limit to retransmit timeout (1 sec. to 60 sec.),the default value is 600.
<max_close_delay> Closing delay required by RFC 793 (100 ms to 7500 ms), the default value is 75.
<is_nack_ind_req> NACK/ACK TCP indication feature. Activating this parameter enables Modem to report the user, in case of losing a TCP connection, what data was received by the remote TCP layer.
0 - feature inactive.
1 - NACK indication active.

2 - ACK indication active.
This parameter resets after power cycle.The default value is 0.

Example:

AT+MIPCONF=1,1,5,20,12,2 //parameters configure success
OK

3.10.2 query command

Show the parameters of TCP/IP stack of every socket.
Syntax:
AT+ MIPCONF?
Response:
+MIPCONF :
<Socket_ID>,<retr_num>,<min_TO>,<max_TO>,<max_close_delay>,<is_nack_ind_req><CR><LF> For all valid sockets.
OK

Example:

AT+ MIPCONF ?
+MIPCONF: 1,1,5,20,12,2
+MIPCONF: 2,4,5,600,75,0
+MIPCONF: 3,4,5,600,75,0
+MIPCONF: 4,4,5,600,75,0
OK

3.10.3 test command

Show the format and available parameters of this command.
Syntax:
AT+ MIPCONF=?
Response:
+MIPCONF:<Socket_ID>,<retr_num>,<min_TO>,<max_TO>,<max_close_delay>,<is_nack_ind_req><CR><LF>
OK

Example:

AT+ MIPCONF=?

+MIPCONF: (1-4),(1-5),(5-10),(10-600),(1-75),(0-2)
OK

3.11 AT+MPING

3.11.1 set command

This command allows to verify IP connectivity to another remote machine (computer) by sending one or more ICMP Echo Request messages. The receipt of corresponding Echo Reply messages are displayed, along with round trip times.Valid IP address must be obtained using AT+MIPCALL command prior to starting ping execution.
NOTE:"MPING" is the unsolicited response. The receipt of corresponding ICMP Echo Reply MPING will be displayed within unsolicited responses, along with round trip times.
NOTE:"MPINGSTAT" is the unsolicited response. when ping request execution is completed, MPINGSTAT provides summary statistics of ping request.
Syntax:
AT+MPING=<mode>[,<"Destination_IP/hostname">[,<count>[,<size>[,<TTL>[,<TOS>[,<TimeOut>]]]]]]
Response:
OK
+MPING:<"Destination_IP">,<type>,<code>,[,<RTT>]
+MPINGSTAT:<status>[,<"Destination_IP">,<SentMessages>,<ReceivedMessages>[,<AverageRTT>]]

<Parameter> Description
<mode>
0 - Abort current ping request execution.
1 - Launch new ping request.
There is no default value - appropriate ERROR will be displayed if parameter is not supplied.
<"Destination_IP/hostname">
Specifies the target machine (computer), which is identified either by IP address 4 octets long in dotted decimal notation or by host name of maximum 255 (not including double quotes) characters long in dotted notation. Each octet of IP address has valid value range of 0 to 255. Host names are not case sensitive and can contain alphabetic or numeric letters or the hyphen. There is no default value - appropriate ERROR will be displayed if parameter is not supplied.
<count> Specifies a number of Internet Control Message Protocol (ICMP) Echo Request messages to send. Valid value range is from 1 to 255.Default value: 4
<TTL> Specifies the length, in bytes, of the Data field in the Echo Request messages sent. The minimum size is 0.The maximum size is 1460.Default value: 32
<TOS> The Type Of Service (TOS) is for internet service quality selection. The type of service is specified along the abstract parameters precedence, delay, throughput, and reliability. These abstract parameters are to be mapped into the actual service parameters of the particular

networks the datagram traverses. Minimum and maximum values for TOS are 0 and 255 respectively. Refer to RFC 791 and RFC 2474 which obsoletes RFC 791 for TOS defined values.Default value: 0

<TimeOut> Specifies the amount of time, in milliseconds, to wait for the Echo Reply message that corresponds to a sent Echo Request message, measured after Echo Request message was sent. If the Echo Reply message is not received within the time-out, +MPINGSTAT

<"Destination_IP"> Specifies the message sender machine (computer),which is identified by IP address 4 octets long in dotted decimal notation. Each octet of IP address has valid value range of 0 to 255. The message sender machine (computer) may be either the target of Echo Requestmessage (if a response was an Echo Reply message) or a gateway (router) in a path of Echo Request message passage for any other ICMP response message.

<type> The first octet of the ICMP header is a ICMP type field, which specifies the format of the ICMP message. Refer to IETF RFC 792 for <type> valid values.

<code> The reasons for the non-delivery of a packet are described by code field value of ICMP header. Every <type> has its own defined <code> values. Refer to IETF RFC 792 for <code> valid values

<RTT> Specifies Round Trip Time (RTT) measured in milliseconds. This parameter will be reported in command response only if Echo Reply message was received.

<status> Specifies a status of ping request execution.

Defined values:

0 - The unsolicited response with this <status> will be sent to DTE upon completion of ping request. If ping request was aborted or socket connection was terminated for any reason, this unsolicited
response will not be reported to DTE.

1 - The unsolicited response with this <status> will be sent to DTE if no ICMP reply message was received within timeout.

2 - The unsolicited response with this <status> will be sent to DTE if socket connection was terminated for any reason. This status essentially means that ping request execution was aborted.

3 - Flow Control OFF. The unsolicited response with this <status> will be sent to DTE if phone doesn't have enough memory to process sending an Echo Request message.

4 - Flow Control ON. The unsolicited response with this <status>
will be sent to DTE if phone has enough memory to send an Echo Request message after flow control was OFF.

<"Destination_IP">

Specifies the target machine (computer) for ping request, which is identified by IP address 4 octets long in dotted decimal notation. Each octet of IP address has valid value range of 0 to 255.

<SentMessages> Specifies a total number of sent Echo Request messages.

<ReceivedMessages> Specifies a total number of received Echo Reply messages corresponding to Echo Request messages.

<AvarageRTT> Specifies average Round Trip Time (RTT) for this ping request.

This value will be reported if and only if <ReceivedMessages> value is greater than zero. Calculation of this value comprises of accumulating all RTT values and dividing total

accumulated RTT by <ReceivedMessages> value. Only an integral part of a result will be reported and any digits of a fraction part will be truncated.

Example:

---

at+mping=1,"www.sina.com"
OK
+MPING: "218.30.108.190",8,0,687
+MPING: "218.30.108.190",8,0,696
+MPING: "218.30.108.190",8,0,655
+MPING: "218.30.108.190",8,0,793
+MPINGSTAT: 0,"218.30.108.190",4,4,707 //ping request 4 times and succeed every time
at+mping=1,"1.1.1.1"
OK
+MPINGSTAT: 1
+MPINGSTAT: 1
+MPINGSTAT: 1
+MPINGSTAT: 1
+MPINGSTAT: 0,"1.1.1.1",4,0 //ping request fail
at+mping=1,"www.tom.com"
OK
+MPING: "202.108.12.68",8,0,424
+MPING: "202.108.12.68",8,0,535
+MPINGSTAT: 1
+MPING: "202.108.12.68",8,0,683
+MPINGSTAT: 0,"202.108.12.68",4,3,547 //ping request 4 times and succeed in 3 times

---

3.11.2 query command

Show the current parameters of ping execution

Syntax:

AT+MPING?

Response:

+MPING:<count>,<size>,<TTL>,<TOS>,<TimeOut>

OK

Example:

AT+MPING?
+MPING: 4,32,255,0,1000
OK

3.11.3 test command

Show the format and available parameters of this command.

Syntax:

AT+MPING=?

Response:

+MPING:<count>,<size>,<TTL>,<TOS>,<TimeOut>

OK

Example:

AT+MPING=?

+MPING: (0-1),(1-255),(0-1372),(1-255),(0-255),(500-600000)

OK

3.12 AT+MSDNS

3.12.1 set command

This command set/read DNS (Domain Name Server) IP address (primary/secondary) for each socket. If the user doesn't specify DNS servers by AT+MSDNS, the Modem will use default DNS from NW. The defined value(s) will be saved during disconnect PDP context (can be used in next PDP context), but will reset after power cycle.

Syntax:

AT+MSDNS=[<Socket_ID>[,<Primary_DNS_server_IP>[,<Secondary_DNS_server_IP>]]]

Response:

OK

<Parameter> Description

<Socket_ID> A unique number that identifies a connection (provided by the terminal application).

0 - Invalid socket number

1,2,3,4 - Valid socket number

5 - Valid socket number dedicated to +MPING.

<Primary_DNS_server_IP>

<Secondary_DNS_server_IP>

IP of the destination site in the format "AAA.BBB.CCC.DDD".

The range of each octant is 0-255. The value can be written in 1, 2, or 3 digits.

Example:

AT+ MSDNS =1,”1.2.3.4”,”4.3.2.1”
OK //Set DNS success

AT+ MSDNS?
+MSDNS: 1,"1.2.3.4","4.3.2.1"
+MSDNS: 2,"211.136.112.50","211.136.20.203"
+MSDNS: 3,"211.136.112.50","211.136.20.203"
+MSDNS: 4,"211.136.112.50","211.136.20.203"
+MSDNS: 5,"211.136.112.50","211.136.20.203"
OK

3.12.2 query command

Show DNS (Domain Name Server) IP address of each socket
Syntax:
AT+ MSDNS ?
Response:
+ MSDNS:
<Socket_ID>,<Primary_DNS_server_IP>,<Secondary_DNS_server_IP><CR><LF> For all valid
sockets.
OK

Example:

AT+ MSDNS ?
+MSDNS: 1,"1.2.3.4","4.3.2.1"
+MSDNS: 2,"211.136.112.50","211.136.20.203"
+MSDNS: 3,"211.136.112.50","211.136.20.203"
+MSDNS: 4,"211.136.112.50","211.136.20.203"
+MSDNS: 5,"211.136.112.50","211.136.20.203"
OK

3.12.3 test command

Show the format of this command.
Syntax:
AT+ MSDNS =?
Response:
+MSDNS: (List of supported <Socket_id>s),(<IP>),(<IP>)
OK

Example:

AT+ MSDNS=?

+MSDNS: (1-5),"(<IP>)","(<IP>)"

OK

3.13 +MIPRTCP

This URC is sent by the MODEM to the terminal when data is received from the TCP protocol stack.

Syntax:

+MIPRTCP: <socket_ID>,<Length>,<CR>,<LF>,<Data>,<CR>,<LF>

<Parameter> Description

<Socket_ID> 1,2,3,4 - Number of valid sockets.

<Length> Size of received Data

<Data> Data received

3.14 +MIPRUDP

This URC is sent by the MODEM to the terminal when data is received from the UDP protocol stack.

Syntax:

+MIPRUDP: <socket_ID>,<Length>,<CR>,<LF>,<Data>,<CR>,<LF>

<Parameter> Description

<Socket_ID> 1,2,3,4 - Number of valid sockets

<Length> Size of received Data

<Data> Data received

3.15 +MIPSTAT

This URC is sent to the terminal indicating a change in status. Currently there are two possible sources of failure, a broken logical connection or a broken physical connection.

Syntax:

+MIPSTAT: <socket_ID>,<n>[,<number_of_acknowled_bytes >]

<Parameter> Description
<Socket_ID> A unique number that identifies a connection.
Valid socket numbers - 1, 2, 3 and 4
<n>
0 - ACK indication
1 - Broken protocol stack
2 - Connection closed automatically due to non-fatal alert
<number_of_acknowled_bytes > Total number of bytes that were acknowledged

4 More Example

4.1 Start a TCP connection

+XDRVI: 9,1,5

+CPIN: READY

+PBREADY

AT+CREG?
+CREG:0,1 //registered to the network
OK

AT+MIPCALL=1,"CMNET"
OK
+MIPCALL: 1 // PDP activated successfully

AT+MIPOPEN=1,5000,"www.sina.com.cn",80,0
+MIPOPEN: 1,1 //socket connect success
or
+MIPSTAT:1,1 //socket connect fail
OK

AT+MIPSETS=1,5 //set buffer autopush size of socket 1
+MIPSETS: 0
OK

AT+MIPSEND=1,"3031323334"//send 5 byte data into buffer of socket 1
+MIPPUSH: 1,0,5 //auto push
+MIPSEND: 1,0,1460 // free size of buffer is 1460
OK

AT+MIPTSEND=1 //send data transparently on socket 1
>Hello<ctrl+z> //maximum length is 1460
+MIPPUSH: 1,0,10 //total 10 bytes is pushed to TCP/IP stack
OK

//If the parameter **is_nack_ind_req** is set to **2** with AT+MIPCONF before start the socket
//connection, following URC appears
+MIPSTAT: 1,0,10 //total 10 bytes is received by remote host

//If data received from the remote host
+MIPRTCP: 1,17,
Hello from server

AT+MIPCLOSE=1 //close the socket connection 1
+MIPCLOSE:1,0
OK

//If connection is terminated by remote server or broken, following URC appears
+MIPSTAT: 1,1

4.2 Disconnection and Retransmission

TCP provides reliable transmission through the use of re transmission of unacknowledged data. The module behaves in the following manner to determine when this should happen.

1: If the link at some point drops out i.e. server crashes, RF coverage is lost, etc, the unit will start retransmission.
2: The socket is closed after the appropriate number of retransmissions has been reached as determined by the AT+MIPCONF command, and the URC +MIPSTAT appears indicating the link is dropped.

Obviously this does not apply to UDP as it is connectionless and unreliable.